IN1010 V21, Obligatorisk oppgave 5

Innleveringsfrist: Mandag 05.04 kl 23:59

Finne sykdomsmønstre i immunrepertoarer

Introduksjon:

I denne oppgaven skal du skrive et program som ved å analysere blodprøver finner sekvensmønstre som indikerer en infeksjon av et bestemt virus. For å få til dette vil vi analysere immunrepertoarene - DNA-sekvenser av immunceller i blodet - til personer som vi vet er blitt smittet eller ikke er blitt smittet av et bestemt virus.

Et immunrepertoar består av mange immunreseptorer som er proteiner som gjenkjenner viruset. Immunrepertoaret til en person er representert av én fil, og hver immunreseptor er representert som en sekvens av (store) bokstaver, en sekvens per linje i filen. Vi ønsker å analysere og finne mønstre i disse sekvensene fra folk som har og som ikke har hatt viruset. Mønstrene vi ser etter er subsekvenser av reseptorsekvensene. Til slutt, etter å ha funnet antall forekomster av alle subsekvensene, vil vi avgjøre hvilke mønstre som i sterkest grad forekommer fortrinnsvis hos personer som vi vet har hatt viruset. Disse dominante mønstrene kan deretter brukes til å diagnostisere nye personer mistenkt for å være smittet av det bestemte viruset. Oppgaven gjenspeiler en tilnærming som i prinsippet kan brukes til å diagnostisere infeksjon av virus som SARS-CoV-2, samt potensielt forbedre diagnostiseringen av autoimmune sykdomer og kreft.

Oppgavebeskrivelse:

Programmet ditt skal lese N filer, der hver fil representerer immunrepertoaret til en person. Hver fil inneholder en rekke linjer eller sekvenser der hver sekvens representerer en immunreseptor. I tillegg er det en metadatafil der hver linje inneholder et filnavn og informasjonen om filen inneholder data som kommer fra en person som har hatt viruset (true) eller kommer fra en person som vi tror ikke har hatt viruset (false).

Programmet skal lete etter subsekvenser av en gitt lengde i immunreseptorene. Denne lengden skal være en konstant i programmet. Vanligvis vil subsekvenslengden være 3 eller 4. Programmet skal først gå gjennom alle filene og for hver fil produsere en mengde av alle subsekvenser av den gitte lengden. Fordi dette er en mengde, vil det ikke være noen duplikater, se eksemplet på slutten.

Men fordi disse mengdene skal behandles videre, skal programmet lagre dem i hashmaper. Nøklene til disse hashmapene skal være en streng med subsekvensen, verdiene skal være objekter som skal inneholde to instansvariabler: subsekvensen og et antall som er én i utgangspunktet. Det vil være én selv om en fil inneholder flere forekomster av samme subsekvens.

Med N filer vil det derfor bli opprettet N hashmaper. For raskere å kunne finne subsekvensene, skal hver fil behandles av en tråd. For enkelhets skyld kan du, hvis du ønsker det, opprette en tråd for hver fil. Ønsker du en større utfordring kan du anta at denne jobben skal gjøres av K tråder, der K er mindre enn antall filer. Når en tråd har fullført behandling av en fil og opprettet en hashmap, så

behandler den en ny fil og lager en ny hashmap, etc. Bare når alle filer er ferdig behandlet kan disse trådene avsluttes.

Du skal programmere en beholderklasse som kan inneholde disse hashmapene. Programmet skal opprette to objekter av denne beholderklassen. Den ene beholderen skal innholde alle hashmapene fra filer fra personer som har hatt viruset (filer merket true), den andre fra personer som ikke har hatt viruset (filer merket false). Trådene som leser filene skal lagre hashmapene de produserer i en av de to beholderne.

Du skal programmere enda en trådklasse hvis oppgave er å hente ut to hashmaper fra en av disse beholderene, slå dem sammen og legge resultatet tilbake i beholderen. En sammenslåing utføres ved å legge sammen antall forekomster av like subsekvenser. Dersom en subsekvens går igjen i de to hashmapene som prosesseres, skal altså den resulterende hashmapen ha denne subsekvensen som én av sine nøkler, hvor verdien skal være summen av verdiene knyttet til subsekvensene i de to hashmapene som ble prosessert. Subsekvenser som ikke er like i de to hashmapene overføres uforandret til resultatet. Først når det bare er én hashmap igjen i hver beholder terminerer disse trådene.

Antall tråder av den siste klassen skal være en parameter til programmet, der for eksempel halvparten behandler den ene beholderen og halvparten den andre.

Det anbefales å gjøre ukesoppgaven for uke 8 før man starter på denne oppgaven. Andre relevante oppgaver er: Triks oppgaver for oblig 5.

Til slutt skal du enten utføre en enkel eller en mer avansert statistisk test:

Enkel "statistisk" test:

Programmet ditt skal avslutte med å finne det vi kaller dominante subsekvenser. For å gjøre det enkelt skal du bare skrive ut alle subsekvenser der antall forekomster hos personer som har hatt viruset minus antall forekomster hos personer som ikke har hatt viruset er større enn eller lik 5. For hver subsekvens skriv også ut de to antallene og denne forskjellen.

Hvis du vil kan du programmere en bedre statistisk test:

Bruk en binomial test for å beregne hvilke mønstre som er dominant hos personer som har hatt viruset. Den binomial testen skal utføres for hver subsekvens. Den inkluderer følgende parametere:

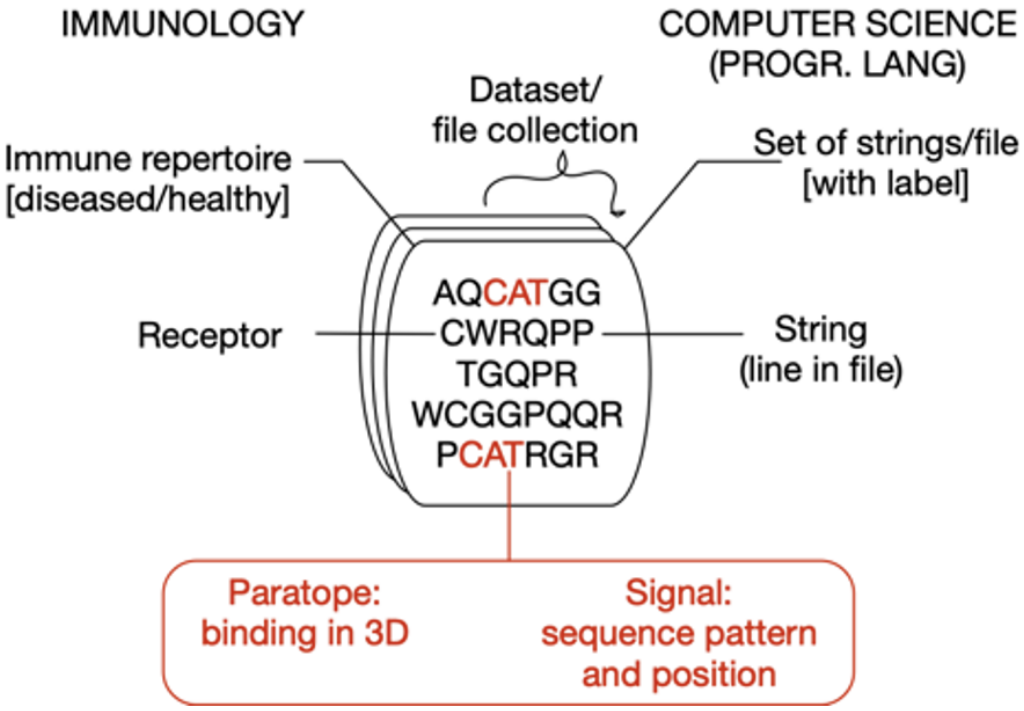
- Antall forsøk: Dette er det totale antallet repertoarer subsekvensen finnes i, som kommer fra både personer som hadde viruset og folk som ikke hadde.
- Antall suksesser: Dette er antallet i repertoarer som kommer fra personer som hadde viruset.
- Sannsynligheten for suksess: Hvor sannsynlig det er for subsekvensen å forekomme hos syke mennesker. Denne settes til 0,5, noe som betyr at det er like sannsynlig å forekomme som å ikke forekomme.
- Typen hypotese som evalueres: Her ser vi bare på subsekvenser som er mer sannsynlig å forekomme hos syke mennesker, så en ensidige test kan brukes her.

Testen gir en p-verdi, og bare subsekvenser med en p-verdi under en terskelverdi (som kan være 0,05) vil bli valgt som dominante mønstre. Terskelen kan være en konstant i programmet.

En implementasjon av en binomial test er tilgjengelig i Apache Commons Math biblioteket, se denne linken for referanse. **Anbefales å lese denne guiden dersom du ønsker å lage en binomial test: guide.**

Når alle subsekvenser er testet, skal programmet avslutte med å skrive ut de dominante mønstrene.

Eksempel på veldig enkelt immunreportoar:



I dette eksemplet inneholder én fil 5 sekvenser. Vi bruker subsekvenser av lengde 3:

AQCATGG 7 bokstaver 5 subsekvenser

CWRQPP 6 bokstaver 4 subsekvenser

TGQPR 5 bokstaver 3 subsekvenser

WCGGPQQR 8 bokstaver 6 subsekvenser

PCATRGR 7 bokstaver 5 subsekvenser

Resultatet fra denne personen vil være disse 23 subsekvensene:

AQC QCA CAT ATG TGG CWR WRQ . . . QQR PCA CAT ATR TRG RGR

CAT er den eneste subsekvensen med flere forekomster, så dette vil resultere i en hashmap med 22 oppføringer, hver med antallet én.